C-based Design: High-Level Synthesis with the Vivado HLx Tool

DSP 3

DSP-HLS-ILT (v1.0)

Course Specification

Course Description

This course provides a thorough introduction to the Vivado® High-Level Synthesis (HLS) tool. It covers synthesis strategies, features, improving throughput, area, interface creation, latency, testbench coding, and coding tips. Use the Vivado HLS tool to optimize code for high-speed performance in an embedded environment and download for in-circuit validation.

Level – DSP 3

Course Duration – 2 days

Price – $1,600 or 16 Xilinx Training Credits

Course Part Number – DSP-HLS-ILT

Who Should Attend? – Software and hardware engineers looking to utilize high-level synthesis

Prerequisites

- C, C++, or System C knowledge
- High-level synthesis for software engineers OR high-level synthesis for hardware engineers

Software Tools

- Vivado System Edition 2017.3
- SDx™ development environment 2017.1
- MATLAB R2017a

Hardware

- Architecture: Zynq®-7000 All Programmable SoC and 7 series FPGAs*
- Demo board: Zynq-7000 All Programmable SoC ZC702 or Zed board and Kintex®-7 FPGA KC705 board*

* This course focuses on the Zynq-7000 All Programmable SoC and 7 series FPGA architectures.

** Check with Morgan Advanced Programmable Systems, Inc. for the specifics of the in-class lab board or other customizations.

After completing this comprehensive training, you will have the necessary skills to:

- Enhance productivity using the Vivado HLS tool
- Describe the high-level synthesis flow
- Use the Vivado HLS tool for a first project
- Identify the importance of the testbench
- Use directives to improve performance and area and select RTL interfaces
- Identify common coding pitfalls as well as methods for improving code for RTL/hardware
- Perform system-level integration of IP generated by the Vivado HLS tool
- Describe how to use OpenCV functions in the Vivado HLS tool

Course Outline

Day 1

- Introduction to High-Level Synthesis {Lecture}
- Basics of the Vivado HLS Tool {Lecture, Demo, Lab}
- Design Exploration with Directives {Lecture}
- Vivado HLS Tool Command Line Interface {Lecture, Lab}
- Introduction to HLS UltraFast Design Methodology {Lecture}
- Introduction to I/O Interfaces {Lecture}
- Block-Level Protocols {Lecture, Lab}
- Port-Level Protocols {Lecture, Demo, Lab}
- Port-Level Protocols: AXI4 Interfaces {Lecture, Demo}
- Port-Level Protocols: Memory Interfaces {Lecture, Lab}
- Port-Level Protocols: Bus Protocol {Lecture}
- Pipeline for Performance: PIPELINE {Lecture, Demo, Lab}

Day 2

- Pipeline for Performance: DATAFLOW {Lecture, Lab}
- Optimizing Structures for Performance {Lecture, Demo, Lab}
- Data Pack and Data Dependencies {Lecture}
- Vivado HLS Tool Default Behavior - Latency {Lecture}
- Reduce Latency {Lecture}
- Improving Area {Lecture, Lab}
- Introduction to HLx Design Flow {Lecture, Demo, Lab}
- HLS vs. SDSoC Development Environment Flow {Lecture, Demo}
- Vivado HLS Tool: C Code {Lecture, Lab}
- Hardware Modeling {Lecture}
- OpenCV Libraries {Lecture}
- Pointers {Lecture}

Topic Descriptions

Day 1

- Introduction to High-Level Synthesis – Overview of the High-level Synthesis (HLS), Vivado HLS tool flow, and the verification advantage.
- Basics of the Vivado HLS Tool – Explore the basics of high-level synthesis and the Vivado HLS tool.
- Design Exploration with Directives – Explore different optimization techniques that can improve the design performance.
- Vivado HLS Tool Command Line Interface – Describes the Vivado HLS tool flow in command prompt mode.
- Introduction to HLS UltraFast Design Methodology – Introduces the methodology guidelines covered in this course and the HLS UltraFast Design Methodology steps.
- Introduction to I/O Interfaces – Explains interfaces such as block-level and port-level protocols abstracted by the Vivado HLS tool from the C design.
- Block-Level Protocols – Explains the different types of block-level protocols abstracted by the Vivado HLS tool.
- Port-Level Protocols – Describes the port-level interface protocols abstracted by the Vivado HLS tool from the C design.
- Port-Level Protocols: AXI4 Interfaces – Explains the different AXI interfaces (such as AXI4-Master, AXI4-Lite (Slave) and AXI4-Stream) supported by the Vivado HLS tool.
- Port-Level Protocols: Memory Interfaces – Describes the Memory Interface port-level protocols (such as BRAM, FIFO) abstracted by the Vivado HLS tool from the C design.
- Port-Level Protocols: Bus Protocol – Explains the bus protocol supported by the Vivado HLS tool.
- Pipeline for Performance: PIPELINE – Describes the PIPELINE directive for improving the throughput of a design.

Day 2

- Pipeline for Performance: DATAFLOW – Describes the DATAFLOW directive for improving the throughput of a design by pipelining the functions to executes as soon as possible.
- Optimizing Structures for Performance – Learn the performance limitations caused by arrays in your design. You will also learn some optimization techniques to handle arrays for improving performance.
- Data Pack and Data Dependencies – Learn how to use DATA_PACK and DEPENDENCE directives to overcome the limitations caused by structures and loops in the design.
- Vivado HLS Tool Default Behavior: Latency – Describes the default behavior of the Vivado HLS tool on latency and throughput.
- Reduce Latency – Describes how to optimize the C design to improve latency.

- Improving Area – Describes different methods for improving resource utilization and explains how some of the directives have impact on the area utilization.
- Introduction to HLx Design Flow – Describes the traditional RTL flow versus the Vivado HLx design flow.
- HLS vs. SDSoC Development Environment Flow – Describes the HLS flow versus the SDSoC™ development environment flow.
- Vivado HLS Tool: C Code – Describes the Vivado HLS tool support for the C/C++ languages, as well as arbitrary precision data types.
- Hardware Modeling – Explains hardware modeling with streaming data types and shift register implementation using the ap_shift_reg class.
- OpenCV Libraries – Explains the OpenCV design flow and the Vivado HLS tool support.
- Pointers – Explains the use of pointers in the design and workarounds for some of the limitations.

Register Today

Morgan Advanced Programmable Systems, Inc. (MAPS, Inc.) delivers public and private courses in locations throughout the central US region; including Iowa, Illinois, Kansas, Minnesota, Missouri, Nebraska, North Dakota, South Dakota and Wisconsin.

Visit **morgan-aps.com/training**, for full course schedule and training information.

Morgan
Advanced Programmable Systems

You must have your tuition payment information available when you enroll. We accept credit cards (Visa, MasterCard, or American Express) as well as purchase orders and Xilinx training credits.

Student Cancellation Policy

- Students cancellations received more than 7 days before the first day of class are entitled to a 100% refund. Refunds will be processed within 14 days.
- Student cancellations received less than 7 days before the first day of class are entitled to a 100% credit toward a future class.
- Student cancellations must be sent here.

MAPS Inc. Course Cancellation Policy

- We regret from time to time classes will need to be rescheduled or cancelled.
- In the event of cancellation, live on-line training may be offered as a substitute.
- MAPS may cancel a class up to 7 days before the scheduled start date of the class; all students will be entitled to a 100% refund.
- Under no circumstances is MAPS responsible or liable for travel, lodging or other incidental costs. Please be aware of this cancellation policy when making your arrangements.
- For additional information or to schedule a private class contact us here.